



TITLE:

SET-UP AND ROUTING ALGORITHMS FOR CELLULAR INTERCONNECTION NETWORK

AUTHOR(S):

HARAO, Masateru; LOMTONG, Paisan

CITATION:

HARAO, Masateru ...[et al]. SET-UP AND ROUTING ALGORITHMS FOR CELLULAR INTERCONNECTION NETWORK. 数理解析研究所講究録 1984, 522: 120-130

ISSUE DATE:

1984-05

URL:

<http://hdl.handle.net/2433/98472>

RIGHT:

SET-UP AND ROUTING ALGORITHMS
FOR CELLULAR INTERCONNECTION NETWORK

Masateru HARAO

and

Paisan LOMTONG

Research Institute of Electrical Communication
Tohoku University
980 Sendai, JAPAN

Abstract A cellular interconnection network(CIN) is a regular array of identical switching elements called cells and by which we realize various graphical interconnection patterns among processors. This paper presents the set-up and routing algorithms for a realization procedure of any graphs in three dimensional CIN using graph theoretical approach. The set-up algorithm takes $O(dn \cdot \log(dn))$ time and $O(dn)$ space and the required area of the CIN is $O((dn)^{3/2})$, where d and n are the degree and the order of the given graph. We exploit also a routing algorithm which can be performed in parallel by self-setting method.

1. Introduction

Concerning with the VLSI algorithms, it has been shown that any graph can be embedded in three dimensional medium using $O(n^{3/2})$ volume though in two dimensional case it needs $O(n^2)$ ^(4,9,10). But the concrete realization algorithm is not apparent there. On the other hand various interconnection networks for VLSI based multi-processors have been proposed⁽¹¹⁾ and their switch setting algorithms are devised^(5,8).

We propose in this paper the three dimensional cellular network as such an interconnection network among processors and design an efficient algorithm to realize any given interconnection patterns. From this viewpoint, CIN model is more powerful than other models and we also emphasize that the routing algorithm can be executed by self-routing in the meaning of Benes network⁽⁸⁾. In Section 2 we describe the formal definitions on this model and some related fundamental results. In Section 3 we show an efficient algorithm to embed any graphs in CIN using graph theoretical approach.

2. Formal Framework and Preliminary Results

2.1. Basic definitions

An undirected graph (or simple graph) G is a pair (V, E) , where V is the set of nodes and $E = \{\{u, v\}; u \neq v, u, v \in V\}$ is the set of edges. If E is replaced with an ordered pair $A = \{(u, v); u \neq v, u, v \in V\}$, then $G = (V, A)$ is called a directed graph (or simple digraph). Any such pair (u, v) is call an arc. Especially the graph of the k -dimensional lattice is denoted by $\mathbb{C}_k = (C_k, S_k)$, where C_k is the k -fold Cartesian product of integer set I , i.e., I^k and $S_k = \{(v, v+s_i); v \in V, s_i = (0, \dots, \overset{i}{1}, 0, \dots, 0), i = 1, \dots, k\}$. In this paper we only deal with the case $k = 3$.

2.2. CIN System

The CIN system is a parallelepiped array of two kinds of cells, one is the processing cell called the active cell (AT-cell) which is located in the first layer, and the other is the switching cell (SW-cell).

By setting the switches of the SW-cells suitably for any given graphs we can realize them homeomorphically in the CIN system. The rough sketch of CIN system is illustrated in Fig.1, where its graphs structure is $\mathbb{C}_3 = (I^3, S_3)$. Hereafter we denote it simply $\mathbb{C} = (C, S)$ if no confusion arises and a CIN of $m \times n$ AT-cells is denoted $\mathbb{C}(m \times n)$ if necessary.

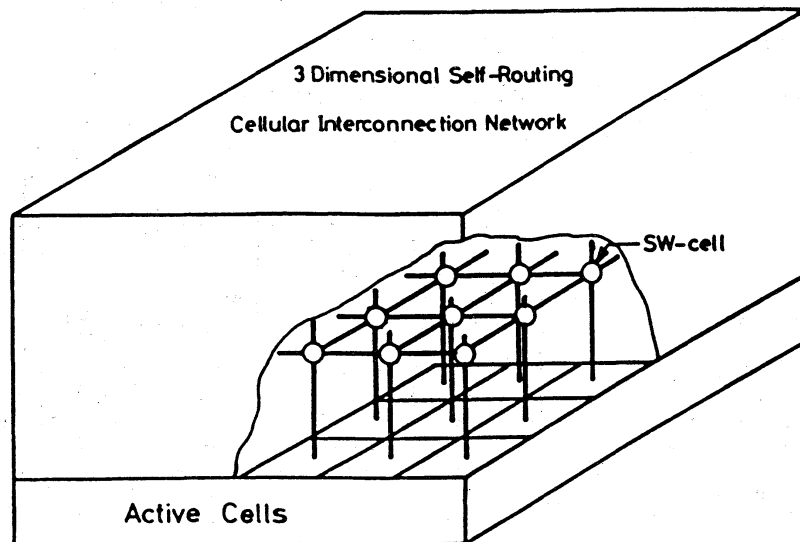


Fig.1. CIN System.

2.3. Realization in CIN

We shall discuss the realization algorithm under the following conditions.

- 1) Each cell can only be interconnected with its direct neighbor cells.
- 2) Duplicated use of the same port is not allowed (no overlapping among the interconnections is allowed).
- 3) Crossings among interconnections are allowed only at the SW-cells.

Definition 1 An embedding (f,g) of $G = (V,E)$ into $C = (C,S)$ is a mapping defined as follows:

- 1) f is an injection from $V \times \{1, \dots, d\}$ into $C \times \{1, \dots, p\}$ called the cell-port assignment, where d and p are the degree of G and the number of ports of AT-cell respectively,
- 2) g maps E into the path of C such that every pair of such paths is edge disjoint.

The procedure to determine a embedding mapping (f,g) is called the set-up algorithm and the realization procedure of the interconnections in CIN using the data processed by the set-up procedure is called the routing algorithm.

3. Realization Algorithm

3.1. Decision of cell-port assignment

We assume that the degree d of the graphs to be embedded in CIN is arbitrary. There are some different organizations for AT-cells and its structure affects the efficiency of the realization (area complexity). Examples of AT-cells embedding degree d graphs are illustrated in Fig.2, where \bullet and x denote the out-port and in-port respectively.

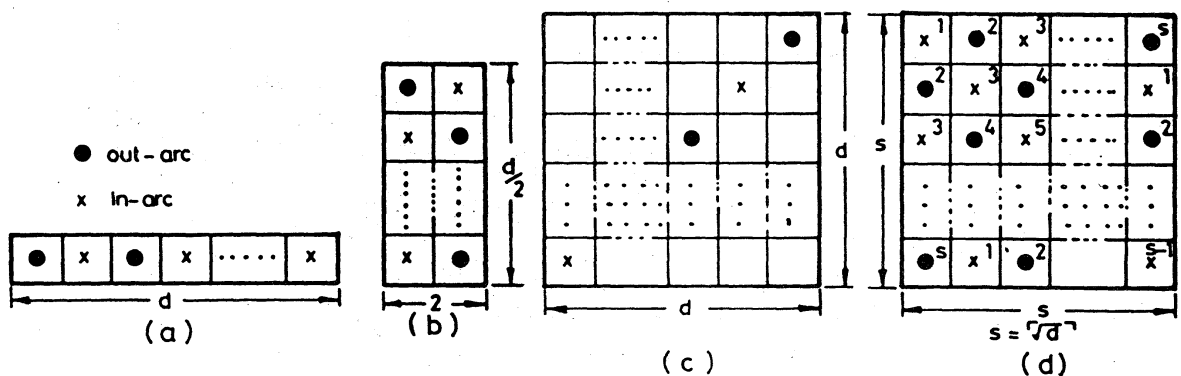


Fig.2. Several examples of AT-cell organization.

The cell-port assignment consists of the following procedures:

- 1) Direction assignment: Transform the given graphs to the directed graphs by assigning directions to each edge.

- 2) Cell assignment: Define the mapping f_1 from V to C .
- 3) Port assignment: Define the mapping f_2 from $v \times \{1, \dots, d\}$ to $f_1(v) \times \{1, \dots, p\}$ for any v .

Proposition 1 Let G be an arbitrary graph of degree d and of order n . Then G can be transformed to a directed graph G^* such that if d is even, then the outdegree and indegree of each node are both $d/2$, and if d is odd, then one of them of each node is $(d+1)/2$ and the other is $(d-1)/2$. The algorithm transforms in $O(dn)$ time and $O(dn)$ space.

Proof: We use the euler partition algorithm⁽²⁾ which partition the edges of a graph into open and closed paths, so each node of odd degree is end of no open paths. According to the direction of eulerian trial, we assign a direction to each edge of G . @

The cell-port assignment is essential to establish an efficient embedding, but it depends on the characteristics of the given graphs. In general we define the mapping $f = (f_1, f_2)$ by the following procedure.

- a) Define an injection f_1 from V into $C = \{(x, y, 0); 0 \leq x, y \leq m\}$, where $m = n^{1/2}$.
- b) Define an injection f_2 from $V \times \{1, \dots, d\}$ into $C \times \{1, \dots, p\}$ as follows;
 - (i) $f_2(u, i)$ is an out-port of $f_1(u)$ if (u, i) is an out-arc of u ,
 - (ii) $f_2(u, i)$ is an in-port of $f_1(u)$ if (u, i) is an in-arc of u .

The cell-port assignment mapping f is the composition of f_1 and f_2 , i.e., (f_1, f_2) defined by $(f_1, f_2)(u, i) = (f_1(u), f_2(u, i))$ for any (u, i) .

3.2. Bipartite graph representation

We assume here that CIN consists of $p \times q$ AT-cells in the first layer and each AT-cell has $s \times t$ ports

Bipartite graph construction

- 1) Partitioning each AT-cell into $s \times t$ smaller segments, we associate the points r_j and c_i to the j -th row and the i -th column, $1 \leq j \leq t \times q$, $1 \leq i \leq s \times p$, respectively as illustrated in Fig.3.
- 2) For a given graph $G = (V, E)$ we transform it to a directed graph $G^* = (V, A)$. We assume that the cell-port assignment mapping $f = (f_1, f_2)$ has already been given. Then the (multi)graph $G_B = (W, B)$ corresponding to $G^* = (V, A)$ under $f = (f_1, f_2)$ is defined as follows;

$$W = \{\{r_j\} \cup \{c_i\}; 1 \leq j \leq t \times q, 1 \leq i \leq s \times p\}$$

$$B = \{\{r_j, c_i\}; f(u, a) \in r_j, f(v, b) \in c_i, (u, v) \in A\},$$

where a, b mean some labels for edges.

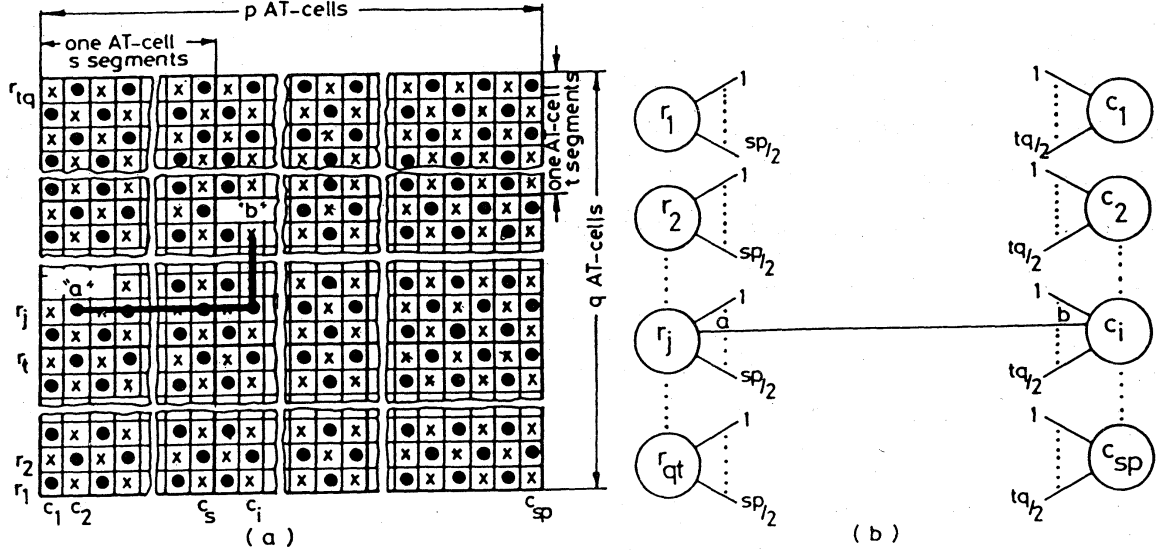


Fig.3. Bipartite graph representation of G .

Proposition 2 The graph $G_B = (W, B)$ is a bipartite graph of order $sp + tq$ and of degree at most $\max(\lceil s/2 \rceil \cdot p, \lceil t/2 \rceil \cdot q)$. (The size of G_B is the same to G)

Proof: From the construction of G_B , the edges exist only from row node to column node. Hence G_B is a bipartite graph. The order is equal the total number of row nodes and column nodes. Since the number of out-port (in-port) of each row and each column of AT-cell are at most $\lceil s/2 \rceil$ and $\lceil t/2 \rceil$ respectively, the total number of edges connected to out-port (in-port) of each row and column of the first layer are at most $\lceil s/2 \rceil \cdot p$ and $\lceil t/2 \rceil \cdot q$ respectively. @

3.3. Layer assignment algorithm

An edge-coloring of a graph G is an assignment of colors to its edges so that no two adjacent edges are assigned the same color. Generally the problem of edge-coloring belongs to the class of NP-complete problems, but for the class of bipartite graphs the more efficient algorithms are available.

We use the modified algorithm of color-by-partition⁽³⁾ and it is noted that the minimum number of required colors is equal to the degree of the bipartite graph.

Lemma 1 Modified color-by-partition algorithm finds a minimum coloring, in time $O(|V|^2 \log |V|)$ and space $O(|E| + |V|)$, where $|V|$ and $|E|$ denote the order and size of G_B respectively.

Procedure color-by-partition⁽³⁾: Comment G is a bipartite graph, all of whose edges are uncolored. A minimum coloring of G is found;

begin

let D be the maximum degree in G ;

if $D = 1$ then color all edges in G , using a new color

else

begin

divide G into edge-disjoint subgraphs G_1 and G_2 having maximum degree D_1 and D_2 , where $D_1, D_2 \leq \lceil D/2 \rceil$ and G_1 has no more edges than G_2 (euler partition⁽²⁾);

color-by-partition (G_1);

remove the edges of r colors from G_1 and add them to G_2 , where $r = 2^{\lceil \log(D/2) \rceil} - D_2$;

euler color⁽²⁾(G_2); Comment now the coloring of G_1 and G_2 give a D -coloring or $D+1$ -coloring of G ;

if G is not D -colored

then

begin

make all edges of some color β uncolored;

for all uncolored edges e do augment⁽³⁾(e);

end

end

end color-by-partition;

As we show in the following part, the embedding of graphs can be established by corresponding each color "b" with the b-th layer.

3.4. Routing procedure

We discuss in this section that how shall we construct the routing mapping g . For each edge adjacent with node r_j assigned the color "b", we realize its connection using b-th layer as illustrated in Fig.4.

Lemma 2 Let G_B be any bipartite graph. Then G_B can be realized in CIN without overlapping by the routing patterns of Fig.4.

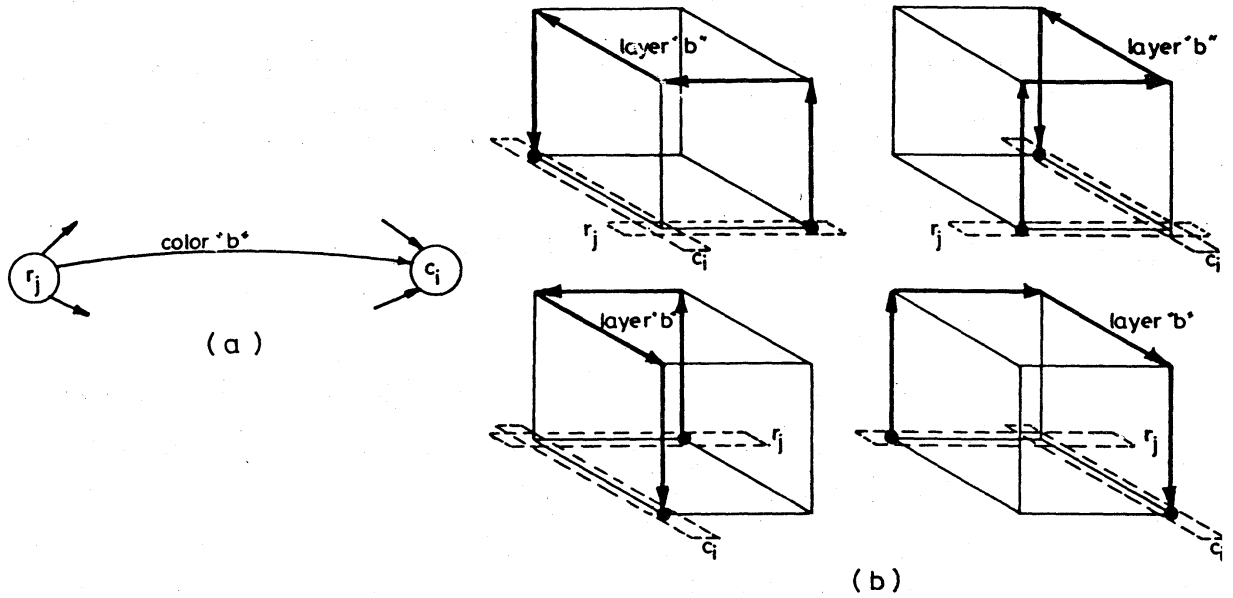


Fig.4. The routing patterns in CIN.

Theorem 1 An arbitrary graph G of degree d and of order n can be realized in CIN using at most $D = \max(\lceil s/2 \rceil \cdot p, \lceil t/2 \rceil \cdot q)$ layers.

Proof: For G , the bipartite graph G_B is of degree $D = \max(\lceil s/2 \rceil \cdot p, \lceil t/2 \rceil \cdot q)$ and of order $V = s \cdot p + t \cdot q$ by Proposition 2. Hence G_B can be D colorable. This implies that G_B can be embedded using D layers. @

Example 1 The degree of the derived bipartite graphs corresponding the AT-cells (a), (b), (c) and (d) of Fig.2 are given as $(sm)/2$, $(sm)/4$, m and $(sm)/2$ respectively. The required area of CIN etc are also given in Table 1.

Table 1.

Cell type	a	b	c	d
Colors(layers)	$\frac{d}{2}m$	$\frac{d}{4}m$	m	$\frac{s}{2}m$
Area of one AT cell	d	d	d^2	d
Area of one layer	dm^2	dm^2	d^2m^2	dm^2
Required SW-cell	$\frac{d^2}{2}m^3$	$\frac{d^2}{4}m^3$	d^2m^3	$\frac{(\sqrt{d}m)^3}{2}$

$$s = \sqrt{d}$$

3.5. Minimal layer realization

From the results of previous section, the number of required layers depends on the organization of AT-cell. In order to establish a minimal layer realization, we have to construct the bipartite graph of minimal degree. As we can observe Table 1, an interesting AT-cell is the one of (d) in Fig.2, which is designed using the property of the Latin squares.

A Latin square is a matrix having integers as its components such that each distinct integer occurs exactly once in each row and each column of it. For designing an AT-cell of d -ports, we use a $s \times s$ Latin square, where $s = \lceil d^{1/2} \rceil$.

Roughly speaking we assign the ports of even number to the out-arcs and the odd ones to the in-arcs. If the number of out-arcs(or in-arcs) of certain nodes exceeds $d/2$, then an extra port not used is to be assigned further(Fig.5). It is noted that the number of out-ports(or in-ports) in each column or each row of the matrix is at most $s/2$, where $s = \lceil d^{1/2} \rceil$.

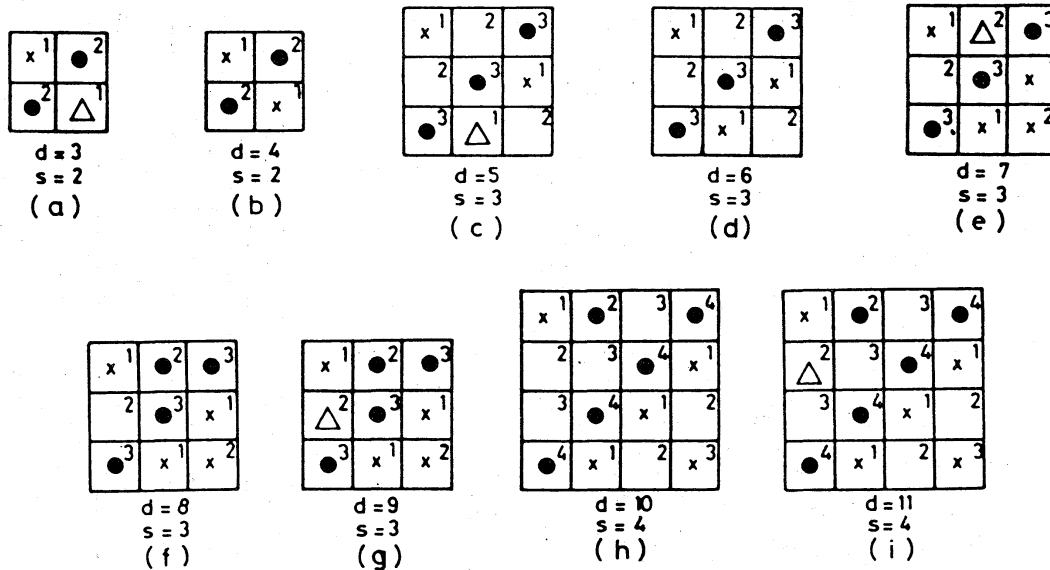


Fig.5. The AT-cell organization based on Latin square.

Theorem 2 (Main theorem) An arbitrary graph G of degree d and of order n can be realized in CIN using $D = (1/2)(dn)^{1/2}$ layers and $(1/2)(dn)^{3/2}$ SW-cells. The realization algorithm runs in $O(dn \cdot \log dn)$ time and $O(dn)$ area.

Proof: From the construction of standard AT-cell we have

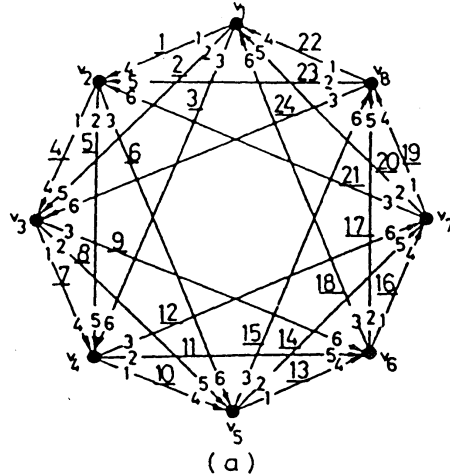
$$s = t = \lceil d^{1/2} \rceil, \quad p = q = \lceil n^{1/2} \rceil.$$

Hence the order, the size and the degree of the derived bipartite graph are given as follows;

$$|V| = 2(dn)^{1/2}, |E| = (dn)/2 \text{ and } |D| = (1/2)(dn)^{1/2}.$$

By applying these relations to Theorem 1 and Lemma 1, we obtain the results. @

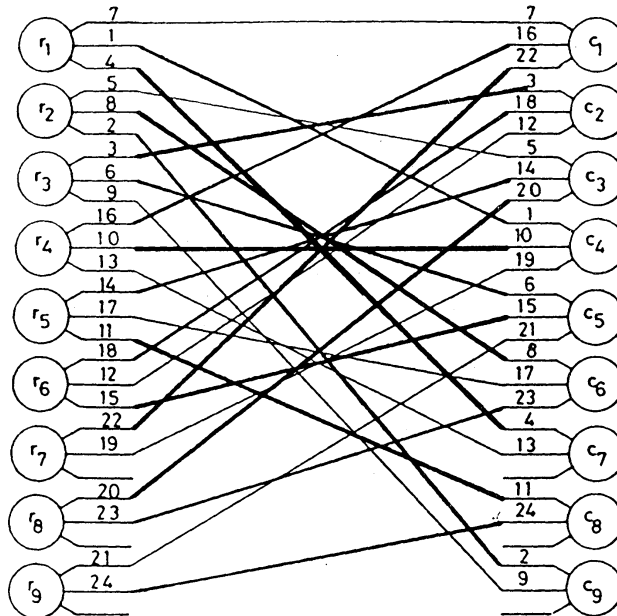
Example 2 Realize the graph in Fig.6 (a) in C.



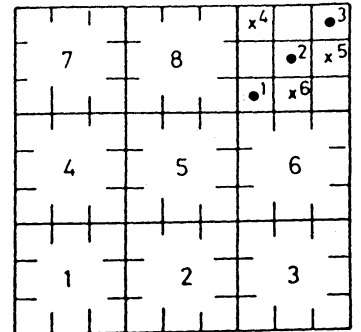
$$h = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ (1,1) & (1,2) & (1,3) & (2,1) & (2,2) & (2,3) & (3,1) & (3,2) \end{pmatrix}$$

$$\beta(u,i) = i \text{ for each } u \text{ and } i, i = 1, 2, \dots, 6.$$

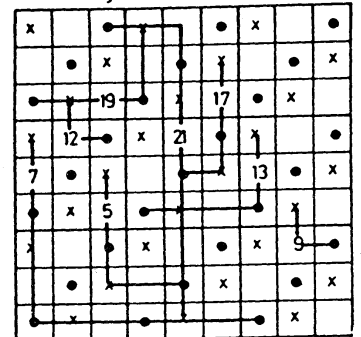
(b)



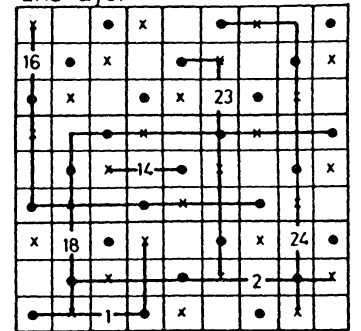
(c)



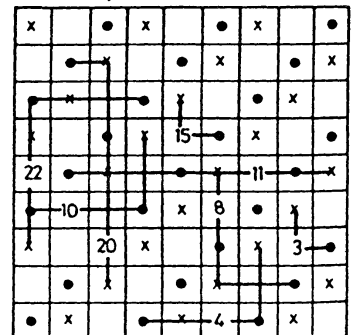
1st layer



2nd layer



3rd layer



(d)

Fig.6. For example 3.

4. Conclusion

As a VLSI based models of interconnection network we proposed the CIN and showed that any graphical interconnection patterns among processors can be realized effectively in it. It seems hopeful to construct some multi-processing system or data flow computing system using interconnection network like CIN⁽¹¹⁾. Further a reconfigurable or programmable array based on this model may be one of the attractive research areas^(6,12). Using this approach a lot of interesting hardware algorithm realizations have been proposed^(1,7,13).

References

- 1) R.W.Floyd and J.D.Ullmann, "The Compilation of Regular Expressions into Integrated Circuits," JACM, vol.29, No.3, 1982, pp. 603-622.
- 2) H.N.Gabow, "Using Euler Partitions to Edge Color Bipartite Multigraphs," Int. J. Comput. and Inf. Science, vol.14, No.4, 1976, pp.345-355.
- 3) H.N.Gabow and O.Kariv, "Algorithm for Edge-coloring Bipartite Graph and Multigraphs," SIAM J. Comput. vol.11, No.1, 1982, pp.117-129.
- 4) K.Hagiwara, N.Suzuki and N.Tokura, "Graph Embedding on a Three Dimensional Model," IECE Trans. vol.J66-D, No.12. 1983, pp.1376-1383.
- 5) G.F.Lev, N.Pippenger and L.G.Valiant, "A Fast Algorithm for Routing in Permutation Networks," IEEE Trans. Comput. vol.C-30, No.2, 1981, pp.93-100.
- 6) P.Lomtong, M.Harao and S.Noguchi, "Cellular Interconnection Arrays for Reconfigurable Computing Systems," IECE, Tech. Commit. of Automaton and Language, EC 83-3, 1983.
- 7) "-----", "Hardware Algorithm Realization with Cellular Reconfigurable Array," IECE, Tech. Commit. of Automaton and Language, AL 83-28, 1983.
- 8) D.Nassimi and S.Sahni, "A Self-routing Benes Network," IEEE Trans. Comput. vol.C-30, 1981, pp.332-340.
- 9) F.P.Preparata, "Optimal Three-Dimensional VLSI Layouts," Mathematical System Theory, vol.16, 1983, pp.1-8.
- 10) A.L.Rosenberg, "Three Dimensional VLSI, A Case Study," IBM T.J.Watson Research Center, Research Report RC 8745, 1981.
- 11) H.J.Siegel, "Interconnection Networks for Parallel and Distributed Processing," Special Issue, IEEE Trans. Comput. vol.C-30, No.4, 1983.
- 12) L.Snyder, "Introduction to the Configurable Highly Parallel Computer,"

IEEE Computer, 1982, pp.47-56.

- 13) S.Wakabayashi, T.Fujii and T.Kikuno,"Pattern Matching Machine for Reconizing Regular Set,"(in Japanese) LA Symp. Record, 1983.